CS 591: Formal Methods in Security and Privacy

Semantics of programs

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

Programming Language

```
c::= abort
   | skip
   | x:=e
   | c;c
   | if e then c else c
   | while e do c
```

x,y,z,... program variables

$e_1, e_2, \ldots$ expressions

$c_1, c_2, \ldots$ commands

Expressions

We want to be able to write complex programs with our language.

```
e::= x
   | f(e1,…,en)
```

Where `f` can be any arbitrary operator.

Some expression examples

`x+5`   `x mod k`   `x[i]`   `(x[i+1] mod 4)+5`

Semantics of Expressions

This is defined on the structure of expressions:

$$\{x\}_m = m(x)$$

$$\{f(e_1, \ldots, e_n)\}_m = \{f\}(\{e_1\}_m, \ldots, \{e_n\}_m)$$

where $\{f\}$ is the semantics associated with the basic operation we are considering.

Semantics of Commands

What is the meaning of the following command?

```
k:=2; z:=x mod k; if z=0 then r:=1 else r:=2
```

We can give the semantics as a relation between command, memories and memories or failure.

$$\texttt{Exp * Mem} \rightarrow \texttt{(Mem + } \perp\texttt{)}$$

We will denote this relation as:

$\{c\}_m = m'$ Or $\{c\}_m = \perp$

This is commonly typeset as:

$$[\![c]\!]_m = m'$$

Semantics of Commands

This is defined on the structure of commands:

$\{\texttt{abort}\}_m = \bot$

$\{\texttt{skip}\}_m = m$

$\{\texttt{x:=e}\}_m = m[x \leftarrow \{\texttt{e}\}_m]$

$\{\texttt{c;c'}\}_m = \{\texttt{c'}\}_{m'}$ If $\{\texttt{c}\}_m = m'$

$\{\texttt{c;c'}\}_m = \bot$ If $\{\texttt{c}\}_m = \bot$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_t\}_m$ If $\{\texttt{e}\}_m = \texttt{true}$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_f\}_m$ If $\{\texttt{e}\}_m = \texttt{false}$

Semantics of While

If $\{e\}_m = \texttt{false}$ Then

$\{\texttt{while e do c}\}_m = m$

What about when $\{e\}_m = \texttt{true}$?

Semantics of While

If $\{\texttt{e}\}_m = \texttt{true}$ Then we would like to have:

$$\{\texttt{while e do c}\}_m = \{\texttt{c;while e do c}\}_m$$

$\boxed{\text{Is this well defined?}}$

Today: more on semantics of While

Well defined?

What is the semantics of the following program:

```
n:=2;
r:=1;
while n ≥ 1 do
  r := n * r;
  n := n-1;
```

$\{\texttt{while e do c}\}_m = \{\texttt{c;while e do c}\}_m$

Approximating While

We could define the following syntactic approximations of a While statement:

$$\texttt{while}^n \; \texttt{e do c}$$

Approximating While

We could define the following syntactic approximations of a While statement:

$$\texttt{while}^{n}\ \texttt{e do c}$$

This can be defined inductively on n as:

$$\texttt{while}^{0}\ \texttt{e do c = skip}$$

$$\texttt{while}^{n+1}\ \texttt{e do c =}$$
$$\texttt{if e then (c;while}^{n}\ \texttt{e do c) else skip}$$

Semantics of While

We could go back and try to define the semantics using the approximations:

$$\{\texttt{while e do c}\}_m = \{\texttt{while}^n\ \texttt{e do c}\}_m$$

Semantics of While

We could go back and try to define the semantics using the approximations:

$$\{\texttt{while e do c}\}_m = \{\texttt{while}^n\texttt{ e do c}\}_m$$

How do we find the n?

Information order

Dana Scott

An idea that has been developed to solve this problem is the idea of information order.

This corresponds to the idea of order different possible denotations in term of the information they provide.

In our case we can use the following order on possible outputs:

$m_1 \quad m_2 \quad m_3 \quad \ldots \quad m_n \quad \ldots$

$\geq \quad \geq \quad \geq \quad \geq$

$\perp$

Semantics of While

Using fixpoint theorems on lattices we can try now to define the semantics using the approximations and a sup operation:

$$\{\texttt{while e do c}\}_m = \sup_{n \in Nat} \{\texttt{while}^n\ \texttt{e do c}\}_m$$

Semantics of While

Using fixpoint theorems on lattices we can try now to define the semantics using the approximations and a sup operation:

$$\{\texttt{while e do c}\}_{\texttt{m}} = \texttt{sup}_{\texttt{n}\in\texttt{Nat}}\{\texttt{while}^{\texttt{n}}\ \texttt{e do c}\}_{\texttt{m}}$$

$\boxed{\text{Will this work?}}$

Semantics of While

Using fixpoint theorems on lattices we can try now to define the semantics using the approximations and a sup operation:

$$\{\texttt{while e do c}\}_m = \sup_{n \in Nat} \{\texttt{while}^n \texttt{ e do c}\}_m$$

Will this work?

We are missing the base case.

Approximating While Revisited

We could define the following lower iteration of a While statement:

$$\texttt{while}_n\ \texttt{e do c}$$

This can be defined using the approximations as:

```
whileₙ e do c =
    whileⁿ e do c;if e then abort else skip
```

Semantics of While

We now have all the components to define the semantics of while:

$$\{\texttt{while e do c}\}_{\texttt{m}} = \texttt{sup}_{\texttt{n}\in\texttt{Nat}}\{\texttt{while}_{\texttt{n}}\ \texttt{e do c}\}_{\texttt{m}}$$

Semantics of Commands

This is defined on the structure of commands:

$\{\texttt{abort}\}_m = \bot$

$\{\texttt{skip}\}_m = m$

$\{\texttt{x:=e}\}_m = m[x \leftarrow \{e\}_m]$

$\{\texttt{c;c'}\}_m = \{\texttt{c'}\}_{m'}$ If $\{\texttt{c}\}_m = m'$

$\{\texttt{c;c'}\}_m = \bot$ If $\{\texttt{c}\}_m = \bot$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_t\}_m$ If $\{e\}_m = \texttt{true}$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_f\}_m$ If $\{e\}_m = \texttt{false}$

$\{\texttt{while e do c}\}_m = \sup_{n \in Nat}\{\texttt{while}_n \texttt{ e do c}\}_m$

where

$\texttt{while}_n \texttt{ e do c} = \texttt{while}^n \texttt{ e do c;if e then abort else skip}$

and

Semantics of Commands

This is defined on the structure of commands:

$\{\texttt{abort}\}_m = \perp$

$\{\texttt{skip}\}_m = m$

$\{\texttt{x:=e}\}_m = m[x \leftarrow \{e\}_m]$

$\{\texttt{c;c'}\}_m = \{\texttt{c'}\}_{m'}$ If $\{\texttt{c}\}_m = m'$

$\{\texttt{c;c'}\}_m = \perp$ If $\{\texttt{c}\}_m = \perp$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_t\}_m$ If $\{e\}_m = \texttt{true}$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_f\}_m$ If $\{e\}_m = \texttt{false}$

$\{\texttt{while e do c}\}_m = \sup_{n \in Nat}\{\texttt{while}_n \texttt{ e do c}\}_m$

where

$\texttt{while}_n \texttt{ e do c} = \texttt{while}^n \texttt{ e do c;if e then abort else skip}$

and

$\texttt{while}^0 \texttt{ e do c} = \texttt{skip}$

$\texttt{while}^{n+1} \texttt{ e do c} = \texttt{if e then (c;while}^n \texttt{ e do c) else skip}$

Example

What is the semantics of the following program:

```
n:=2;
r:=1;
while n ≥ 1 do
  r := n * r;
  n := n-1;
```

Example

What is the semantics of the following program:

```
Fact(n: Nat) : Nat
  r:=1;
  while n > 1 do
     r := n * r;
     n := n-1;
```

Hoare Triples

Hoare triple

Precondition
(a logical formula)

Precondition
Program
Postcondition

$$c : P \Rightarrow Q$$

Program

Postcondition
(a logical formula)

Some examples

$$x = z + 1 : \{z > 0\} \Rightarrow \{x > 1\}$$

Some examples

$$x = z + 1 : \{z > 0\} \Rightarrow \{x > 1\}$$

Is it valid?

Some examples

$$x = z + 1 : \{z > 0\} \Rightarrow \{x > 0\}$$

Some examples

$$x = z + 1 : \{z > 0\} \Rightarrow \{x > 0\}$$

Is it valid?

Some examples

$$x = z + 1 : \{z < 0\} \Rightarrow \{x < 0\}$$

Some examples

$$x = z + 1 : \{z < 0\} \Rightarrow \{x < 0\}$$

Is it valid?

Some examples

$$x = z + 1 : \{z = n\} \Rightarrow \{x = n + 1\}$$

Some examples

$$x = z + 1 : \{z = n\} \Rightarrow \{x = n + 1\}$$

Is it valid?

Some examples

```
while x>0
    z=x*2+y
    x=x/2
    z=x*2-y
```

$$: \{y > x\} \Rightarrow \{z < 0\}$$

Some examples

```
while x>0
    z=x*2+y
    x=x/2
    z=x*2-y
```

$$: \{y > x\} \Rightarrow \{z < 0\}$$

Is it valid?

Some examples

```
while x>0
    z=x*2+y
    x=x/2
    z=x*2-y
```

$$: \{y > x\} \Rightarrow \{z < 0\}$$

Some examples

```
while x>0
    z=x*2+y
    x=x/2
    z=x*2-y
```

$$: \{y > x\} \Rightarrow \{z < 0\}$$

Is it valid?

Some examples

```
 z=x*2+y
x=x/2
 z=x*2-y
```

$: \{\text{even}\, y \wedge \text{odd}\, x\} \Rightarrow \{z < \sqrt{2.5}\}$

Some examples

```
z=x*2+y
x=x/2
z=x*2-y
```

$: \{\text{even}\, y \wedge \text{odd}\, x\} \Rightarrow \{z < \sqrt{2.5}\}$

Is it valid?

How do we determine the validity of an Hoare triple?

Validity of Hoare triple

Precondition
(a logical formula)

$$c : P \Rightarrow Q$$

Program

Postcondition
(a logical formula)

Validity of Hoare triple

We are interested only in inputs that meets P and we want to have outputs satisfying Q.

Precondition
(a logical formula)

$$c : P \Rightarrow Q$$

Program

Postcondition
(a logical formula)

Validity of Hoare triple

Precondition
(a logical formula)

$$c : P \Rightarrow Q$$

Program

Postcondition
(a logical formula)

We are interested only in inputs that meets P and we want to have outputs satisfying Q.

How shall we formalize this intuition?

Validity of Hoare triple

We say that the triple $\texttt{c:P}\Rightarrow\texttt{Q}$ is valid

if and only if

for every memory $\texttt{m}$ such that $\texttt{P(m)}$

and memory m' such that $\{\texttt{c}\}_{\texttt{m}}\texttt{=m'}$

we have $\texttt{Q(m')}$.

Validity of Hoare triple

We say that the triple $c:P \Rightarrow Q$ is valid

if and only if

for every memory m such that $P(m)$

and memory m' such that $\{c\}_m = m'$

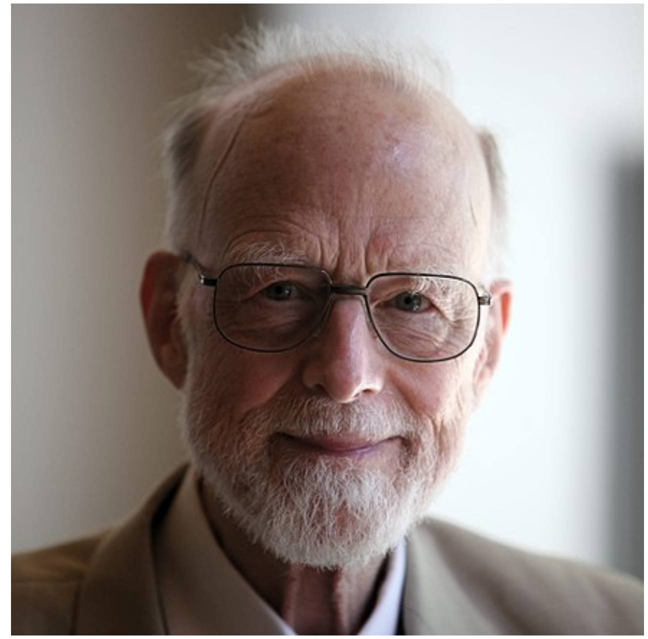
we have $Q(m')$.

Is this condition easy to check?

Hoare Logic

Floyd-Hoare reasoning

Robert W Floyd



Tony Hoare

A *verification* of an interpretation of a flowchart is a proof that for every command c of the flowchart, if control should enter the command by an entrance a_i with P_i true, then control must leave the command, if at all, by an exit b_j with Q_j true. *A semantic definition* of a particular set of command types, then, is a rule for constructing, for any command c of one of these types, a *verification condition* $V_c(\mathbf{P};\mathbf{Q})$ on the antecedents and consequents of c. This verification condition must be so constructed that a proof that the verification condition is satisfied for the antecedents and consequents of each command in a flowchart is a verification of the interpreted flowchart.

Rules of Hoare Logic
Skip

$$\frac{}{\vdash \texttt{skip}:\ P \Rightarrow P}$$

Rules of Hoare Logic
Composition

$$\frac{??}{\vdash \texttt{c;c'}: \quad P \Rightarrow Q}$$

Rules of Hoare Logic
Composition

$$\frac{}{\vdash \texttt{c;c'} : P \Rightarrow Q}$$

Rules of Hoare Logic
Composition

$$\frac{\vdash \texttt{c} : P \Rightarrow R}{\vdash \texttt{c;c'} : \ P \Rightarrow Q}$$

Rules of Hoare Logic
Composition

$$\frac{\vdash \texttt{c} : P \Rightarrow R \qquad \vdash \texttt{c}' : R \Rightarrow Q}{\vdash \texttt{c;c}' : P \Rightarrow Q}$$

Rules of Hoare Logic
If then else

$$\frac{}{\vdash \texttt{if e then } c_1 \texttt{ else } c_2 \texttt{ : } P \Rightarrow Q}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash \texttt{c}_1 : \texttt{P} \Rightarrow \texttt{Q}}{\vdash \texttt{if e then c}_1 \texttt{ else c}_2 : \texttt{P} \Rightarrow \texttt{Q}}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash \mathtt{c}_1 : \mathtt{P} \Rightarrow \mathtt{Q} \qquad\qquad \vdash \mathtt{c}_2 : \mathtt{P} \Rightarrow \mathtt{Q}}{\vdash \mathtt{if\ e\ then\ c}_1\ \mathtt{else\ c}_2 : \mathtt{P} \Rightarrow \mathtt{Q}}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash \texttt{c}_1 : P \Rightarrow Q \qquad \vdash \texttt{c}_2 : P \Rightarrow Q}{\vdash \texttt{if e then c}_1 \texttt{ else c}_2 : P \Rightarrow Q}$$

Is this correct?

Rules of Hoare Logic
If then else

$$\frac{}{\vdash \texttt{if e then } c_1 \texttt{ else } c_2 : P \Rightarrow Q}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash c_1 : e \wedge P \Rightarrow Q}{\vdash \texttt{if } e \texttt{ then } c_1 \texttt{ else } c_2 : P \Rightarrow Q}$$

Rules of Hoare Logic
If then else

$$\frac{\vdash c_1 : e \wedge P \Rightarrow Q \qquad \vdash c_2 : \neg e \wedge P \Rightarrow Q}{\vdash \mathtt{if}\ e\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 : P \Rightarrow Q}$$

Rules of Hoare Logic Assignment

$$\frac{}{\vdash \texttt{x:=e:}\ \ P \Rightarrow P[e/x]}$$

Rules of Hoare Logic
Assignment

$$\vdash \texttt{x:=e:}\quad \texttt{P} \Rightarrow \texttt{P[e/x]}$$

Is this correct?

Rules of Hoare Logic
Assignment

$$\frac{}{\vdash \texttt{x:=e} \;:\; P[e/x] \Rightarrow P}$$

Rules of Hoare Logic
While

$$\frac{}{\vdash \texttt{while e do c : P} \Rightarrow \texttt{P} \wedge \neg \texttt{e}}$$

Rules of Hoare Logic
While

$$\frac{\vdash \texttt{c} : \texttt{e} \wedge \texttt{P} \Rightarrow \texttt{P}}{\vdash \texttt{while e do c} : \texttt{P} \Rightarrow \texttt{P} \wedge \neg\texttt{e}}$$

How do we know that these are the right rules?

Correctness of a rule

$$\frac{}{\vdash \texttt{c} \; : \; \texttt{P} \Rightarrow \texttt{Q}}$$

Correctness of a rule

$$\frac{\vdash c' \;:\; R \Rightarrow S}{\vdash c \;:\; P \Rightarrow Q}$$

Correctness of a rule

$$\frac{\vdash c' : R \Rightarrow S}{\vdash c : P \Rightarrow Q}$$

We say that a rule is correct if given a valid triple as described by the assumption(s), we can prove the validity of the triple in the conclusion.